Vilnius University

Faculty of Mathematics and Informatics

Master's Degree Program of Software Engineering

Course 1, Group 1

Internet Voting System
ICT Feasibility Study
A Guideline for External Audit

Kęstutis Matuliauskas

Version 1.2

January 14th, 2019

Table of contents

1. Goals of analysis

The goal of this analysis and its alternatives is to ensure that the best possible scenario was chosen to reach the goals of the systems development with balance between least efforts, best price and fit to roadmap timeline.

As the system audit won't ever be successful if it is done by internal team member's or documentation author, it has to be done by external auditor (person or company), if needed. The importance of documentation author is to advise and recommend the auditor how he can perform the best results and quality on his audit process. Given advises and recommendation are not required – it is fully upon auditor to accept the, use part of them, or work via his own methods.

2. Non-functional requirements feasibility & coherence analysis for external auditors

2.1. Recommended checklists to external auditors about requirements capture

Checklist for Requirements Capture from Usability perspective:

I. Have you identified all of the system's key touch points?
II. Have you identified all of the different types of users who will interact with the system?
III. Do you understand the type of usage (occasional, regular, transactional, unstructured) for each of the touch points?
IV. Have you taken into account the needs of support and maintenance staff and other second-line users?
V. Do you understand the capabilities, experience, and expertise of the system's users? Have you correctly mapped these into requirements for presentation and support?
VI. Have you taken into account any corporate standards for presentation and interaction, particularly for systems exposed to the public?

2.2. Recommended questions to answer for external auditors during product's feasibility study

In general, the following questions are addressed in the feasibility study of the system:

I. Is it generally known how the requirements can be implemented?
II. Is the knowledge and payments available to the team of executors enough for that?
III. Is there enough money for the project?
IV. Is it enough time for the project?
V. Are there any legal or other obstacles to this?
VI. Will the business really benefit from the actual implementation of the requirements?

2.3. Recommended key aspects about product's feasibility to analyze for external auditors

The feasibility of the system is analyzed in five key aspects:

I. **Operating:** Is the customer able to operate the developed system? Will the expected system usage scenario really work? Do the professionals in the subject matter have an interest in complying with the rules set out in the scenario? The operational feasibility of the system analyzes various obstacles that may prevent it, such as keyboard workloads, computer fears, traditions, corporate culture, and so on.
II. **Technical:** Is there a problem-solving theory and is there supportive technology? Do implementers have the ability to create a system?
III. **Economical:** Will the project pay off? How long does it take to invest in the system?
IV. **Plan:** Is it possible to complete the project on time with the existing executives and other available resources?
V. **Legal Ethical:** Does the project violate any applicable law or any recognized ethical standards?

2.4. Recommendations for audit of non-functional system requirement groups

Auditable non-functional requirement groups:

I. System's performance
II. System's restore after failure
III. System's availability
IV. Voting security
V. Trusted voting
VI. Singularity of vote
VII. Vote verification
VIII. Inspections and analysis for monitoring agencies (i.e. "Transparency international")
IX. Openness (open-source)

2.5. Recommended key risk factors of requirement feasibility to analyze

There are ten key risk factors for the feasibility of the requirement to be analyzed:

I. Requirement variability
II. Impact on system performance
III. Impact on the reliability of the system,
IV. Impact on the security of use of the system and its protection against unauthorized use
V. Changes you may have to make in the current software system engineering process
VI. The need to use unusual technologies for operators
VII. The need to work with non-standard data
VIII. The possibility of violation of project implementation deadlines
IX. The need to hire subcontractors
X. Dependence on third countries

2.6. External audit

External audit about non-functional requirements coherence has to be done by independent external organization, that has technical knowledge of the technology field, which is in this case is the technologies of digital democracy (internet voting, online petitions etc.), paper-voting process and internet voting process standard flow, plus they must have a legal expert on the team, to validate both functional & non-functional requirement against legal law of some of expected buyers countries or regions, i.e. within European Union and US. This audit can also be done by independent team inside the same organization that works on different projects, but, preferred, that those projects would be in the same technology field, which is in this case – systems in digital democracy technology field.

3. Architectural decisions & considered alternatives

3.1. WordPress CMS – Cross-Cutting Architectural Decision

I. A content management system or programming language framework to run the Internet Voting System
   i. Status: accepted
   ii. Deciders: CEO, Software Architect
   iii. Date: Dec 10, 2018

II. Context and Problem Statement

III. Decision Drivers
   i. A need for easy-to-start system
   ii. A need for highly maintainable system
   iii. A system that works for many server platforms, providers

IV. Considered Options
   i. WordPress
   ii. Symfony
   iii. System on .NET

V. Decision Outcome
   i. Chosen option 1 – WordPress, because that is the only easy-to-start, highly maintainable system.

VI. Pros and Cons of the Options
   i. Pros – Will fit and do the job as expected
   ii. Cons – It requires often update

3.2. S.O.L.I.D. MVC – Architectural Decision for Development & Information Views

I. S.O.L.I.D. MVC
   i. Status: accepted
   ii. Deciders: CEO, Software Architect
   iii. Date: Dec 10, 2018

II. Context and Problem Statement

III. Decision Drivers

IV. Considered Options

V. Decision Outcome

VI. Pros and Cons of the Options

3.3. SemVer – Architectural Decision for Development & Information Views

I. Semantic Versioning (SemVer)
   i. Status: accepted
   ii. Deciders: CEO, Software Architect
   iii. Date: Dec 10, 2018

II. Context and Problem Statement
   i. Dependency hell when trying to activate the system, update the system, run beta versions, or process rollback.

III. Decision Drivers
   i. Known pattern to control the versioning.
   ii. Widely-accepted
   iii. Cross-platform, language-independent.

IV. Considered Options
   i. My own versioning system (A.B)
   ii. SemVer

V. Decision Outcome

Chose option 2 (SemVer).

   i. Positive outcome:
   1) Simple upgrades, rollbacks, version comparing – we always know which version is later, which is earlier.

   ii. Negative outcome:
   1) A bit more of work to implement
   2) Not that simple versioning

VI. Pros and Cons of the Options
   1) Option 1 (My own versioning system)

Pros: Code already exist. Can be used from start.

Cons: No way to work with other system, no easy way to notify correctly upgrade service. No support for patching.

   2) Option 2 (SemVer)

Pros: Cross-platform, language-independent, widely accepted.

Cons: Many params on version, not that simple versioning.

4. Findings

From software architect, who is building this system, perspective, all the non-functional requirements are coherent and feasible – that means that there is none non-functional requirement that would be blocking another non-functional requirement to implement the system.

For the external auditors' perspective, only after external audit is completed and results are shared with the seller's company and document author, only then it can be decided, if all non-functional requirements are correctly formulated, can be verified and does not block each other.